Volume : 2 | Issue : 4 | Apr 2013 • ISSN No 2277 - 8179

Research Paper

A Study on Ontology-Based Cloud in Best-Practice Evaluation

Information Technology

KEYWORDS : Cloud computing, ontology, cloud storage

Anandakumar. H

Assistant professor, Department of Information Technology, SNS College of Engineering, Coimbatore-641 107, Tamilnadu, India.

ABSTRACT

Cloud computing is a term used to describe both a platform and type of application. A cloud computing platform forcefully provisions, configures, reconfigures, and de-provisions servers as needed. Cloud Computing is one contemporary technology in which the research community has recently embarked. The unpredictable demands of software and hardware IT infrastructure have resulted in Cloud computing to be the fastest growing fashion in the Information Technology industry. As from industry perspective, organizations espouse cloud computing as they no longer need to buy or maintain expensive and energy-draining equipments. IT management including licensing issues, software updates and IT safety management, all are taken care by the cloud freelancer. Amputation of this administrative trouble allows organizations to concentrate on their core business and be more productive. We propose in this paper ontology of this area which demonstrates a dissection of the cloud into their inter-dependency on preceding technologies.

1. INTRODUCTION

Cloud computing is an emerging computing technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows customers and businesses to use applications without installation and access their personal files at any computer with internet admittance. This technology allows for much more capable computing by centralizing. It allows users to temporary utilize computing infrastructure over the network, supplied as a service by the cloud service provider at possibly one or more levels of a paradigm that allows users to temporary utilize computing infrastructure over the network, supplied as a service by the cloud-service provider at possibly one or more levels of a business models rapidly evolved to harness this technology by providing application software, programming platforms, information-storage, computing infrastructure and hardware as services.

1.1 Limitations of current cloud system

Grid Computing and virtualization are the most obvious predecessor technologies that enabled the advent of cloud computing. However, several other computing concepts have indirectly shaped today's cloud computing technology, including peer-to-peer (P2P) computing, Service Oriented Architecture and autonomic computing. MapReduce is an example of programming models that demonstrated a simpler way to develop data-intensive applications for large distributed systems. A cloud infrastructure can be a cost efficient model for delivering in order services, sinking IT management complexity, promoting novelty, and increasing responsiveness through real time workload balancing. The Cloud makes it possible to start Web 2.0 applications rapidly and to scale up applications as much as needed when needed. Understanding the implications of developing cloud applications against one cloud layer versus another will equip developers with the knowledge to make informed decisions about their applications' expected time-to-market, programming productivity, scaling flexibility as well as feat bottlenecks. In this regard, an ontology can facilitate the adoption of cloud computing.

1.2. Classification Methodology

In order to define ontology of cloud computing, we had to choose a method to classify the different cloud systems. We select to use compos ability as our methodology, as it enables our proposed ontology to detain the inter-relations between the different cloud components. For simplicity, we define our proposed ontology to be envisioned as a stack of layers. Each layer encompasses one or more cloud services. For example, all cloud software environments (a.k.a. cloud platforms) target programmers, though cloud applications target end users. The cloud software environments would be classified in a different layer than cloud applications. By compos ability, we categorize one cloud layer to be higher in the cloud stack, if its services can be collected from the services of the basic layer. Consider the cloud application layer as an example, since we can enlarge cloud applications using cloud software environments, we say that cloud applications are compos able from cloud Application software environments, and that the cloud Application Software layer is higher in the cloud stack.

2. THE CLOUD ONTOLOGY

Cloud computing system fall into five layers:

- Applications
- Software environments
- Software infrastructure
- Software kernel
- Hardware

The individuality and boundaries of each of the layers are given.

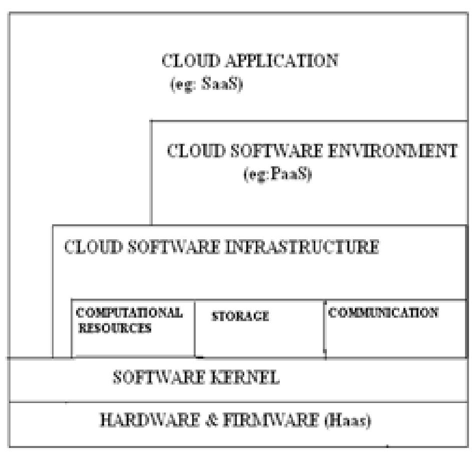

Fig 1: Cloud Computing Ontology-Depicted as Five Layers.

2.1 Application Layer

The cloud application layer is the most visible layer and services provided by this layer through web-portals. Furthermore, it exports the computational work from the users' terminal to data centers where the cloud applications are deployed.

Configuration and testing of the application in this model is arguably less complicated, since the deployment environment becomes restricted, i.e., the provider's data center. This model conveys several favorable benefits for the users and providers of cloud applications, and is usually referred as Software as a Service (SaaS). Furthermore, they become less error-prone since all their interactions with the cloud are through pretested APIs.

Despite all the advantageous benefits of this model, several execution issues hinder its wide adoption. Specifically, the security and ease of use of the cloud applications are two of the major issues in this model, and they are presently avoided by the use of lenient service level agreements (SLA).

Volume : 2 | Issue : 4 | Apr 2013 • ISSN No 2277 - 8179

2.2 Software Environment Layer
The cloud software environment layer is also called the software platform layer. The users of this software application layer are cloud applications developers, implementing the applications and deploying them on the cloud.

The providers of the cloud software application environments supply the developers with a programming-language-level environment with a set of well-defined APIs to facilitate the interaction between the environments and the cloud applications, as well as to go faster the deployment and support the scalability needed of those cloud applications.

The service provided by cloud systems in this layer is commonly referred to as Platform as a Service (PaaS). Example of systems in this category is Google's App Engine. A Hadoop deployment on the cloud would be considered a cloud application software environment, as it provides its developers with a programming environment, i.e. maps reduce framework for the cloud.

Similarly, Yahoo's Pig , a high-level language to enable processing of very large files on the hadoop environment may be viewed as an open-source implementation of the cloud platform layer. As such, cloud application software environments make possible the process of the development of cloud applications.

2.3 Software Infrastructure Layer
The cloud software infrastructure layer provides fundamental resources to other higher-level layers, which in twirl can be used to construct new cloud software environments or cloud applications. Our proposed ontology reflects the fact that the two highest levels in the cloud stack can bypass the cloud infrastructure layer in building their system. Although this bypass can enhance the efficiency of the system, it comes at the cost of simplicity and development efforts. Cloud services offered in this layer can be categorized into:

- Computational resources.
- Data storage.
- Communications.

2.3.1 Computational Resources
Virtual machines (VMs) are the most common form for providing computational resources to cloud users at this layer, where the users get finer-granularity flexibility since they normally get super-user access to their VMs, and can use it to customize the software stack on their VM for performance and efficiency. Often, such services are dubbed Infrastructure as a Service (IaaS). Virtualization is the enabler technology for this cloud component, which allows the users unique flexibility in configuring their settings while protecting the physical infrastructure of the provider's data center. Recent advances in OS Virtualization have made the concept of IaaS plausible. This was specifically enabled by two virtualization technologies:

- Paravirtualization
- Hardware-assisted virtualization.

2.3.2 . Data Storage
The second infrastructure resource is data storage, which allows users to store the data at remote disks and access them anytime from any place. This service is normally known as Data-Storage as a Service (DaaS), and it facilitates cloud applications to balance beyond their limited servers. Data storage systems are predictable to meet several meticulous requirements for maintaining users' data and information, including availability, reliability, performance, replication and data stability but because of the conflicting nature of these requirements, no one system executes all of them together. For example, ease of use, scalability and data stability can be regarded as three conflicting goals. While those features are hard to be met with general data storage systems, DaaS-providers have taken the liberty of implementing their system to favor one feature over the others, while indicating their choice through their SLA. These implementations have borrowed their fundamental ideas from proceeding research and production systems. Some examples of data storage systems are: distributed file systems (e.g., GFS), replicated relational databases (RDBMS) (e.g., Bayou) and key-value stores (e.g., Dynamo).

2.3.3. Communication
As the need for a guaranteed quality of service (QoS) for network communication grows for cloud systems, communication becomes an imperative component of the cloud infrastructure. Accordingly, cloud systems are obliged to provide some communication capability that is service oriented, secluded, configurable, schedulable, expected, and reliable. Towards this objective, the concept of Communication as a Service (CaaS) emerged to support such requirements, as well as network security, forceful provisioning of virtual overlays for traffic isolation or committed bandwidth, guaranteed message delay, communication encryption, and network monitoring. One recent example of systems that belong to CaaS is Microsoft Connected Service Framework (CSF).

- **Software Kernel:**

This cloud layer provides the basic software management for the physical servers that compose the cloud. Software kernels at this point can be implemented as an OS kernel, hypervisor, and virtual machine supervise and clustering middleware. The two most successful grid middleware that harness the physical resources to provide a successful deployment environment for Grid applications were arguably Globus and Condor.

- **Hardware And Firmware:**

The bottom layer of the cloud stack in our proposed ontology is the actual physical hardware and switches that form the backbone of the cloud. The users of this layer of the obscure are normally big enterprises with huge IT requirements in need of subleasing Hardware as a Service (HaaS).

3. CLOUD PROVISIONING AND MANAGEMENT
3.1 Automated Provisioning
The core functionality of a cloud is its ability to automatically provision servers for innovators and to allow innovators, administrators, and others to use that function with a Web-based

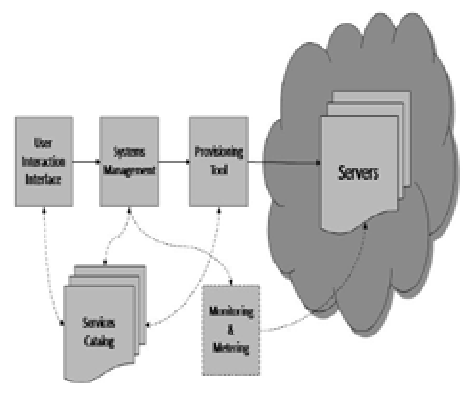

Fig 2: The Cloud System.
Interface. The role-based interface abstracts out the complexity of IBM Tivoli Provisioning Manager, Remote exploitation Manager, Network Installation Manager, business process execution language (BPEL), and Web services. Typically, a pilot team needs four to twelve weeks to identify, procure, and execute a pilot infrastructure and additional time to build a security compliant software stack so that developers can begin building or deploying applications and code. The cloud provides a framework and offering that reduces that boarding process to approximately one hour. We accomplish this through a role-based Web portal that allows innovators to fill out a form defining their hardware platform, CPU, storage, memory, operating system, middleware, and team members and associated roles. The process is fully automatic and completes in about an hour. The value of having a fully automated provisioning process that is security compliant and automatically customized to innovators' needs is manifested in reduced time to introduce technologies and innovations, investments in labor for designing, procuring,

and construct hardware and software platforms, and cost evasion in higher use and reuse of existing resources.

2.2 Reservation And Scheduling

Critical to an environment like a cloud is the ability to understand what your current and future capacity is to occupy the customers. Without that understanding you cannot accurately forecast how many customers can support, nor can you ensure that you maintain a steady pipeline of modernization. For this reason, projects cannot board the cloud without an agreed upon end date. This date which is part of the contract offers an incentive for the project team to work aggressively to meet their target or risk the removal of the resource assigned.

2.3 Extending Contract

As with most modernization, projects, pilots, and prototypes often have unknown elements related to improvement delays, new requirements, and grant. The unknowns sometimes make it difficult to adhere to a contractual end date, particularly if the end date was agreed to several months before the project delays.

An innovator can choose to:

- Delete a server
- Add software to a server
- Reimage a server with a novel operating system
- Change the memory or storage space allocation of a server
- Add a server

4. USAGE SCENARIOS

Here we summarize several basic but important usage scenarios that highlight the breadth and depth of impact that cloud computing can have on an enterprise.

4.1 Internal Innovation

Innovators request resources online through a simple Web interface. They specify a desired start and end dates for their channel. A cloud resource administrator approves or rejects the request. Upon approval, the cloud provisions the servers. The innovator has the resources offered for use within a few minutes or an hour depending on what type of resource was requested.

4.2 Virtual Worlds

Virtual worlds require significant amounts of computing power, particularly as those virtual spaces become large or as more and more users log in. Especially multiplayer online games (MMPOG) are a good example of significantly large virtual worlds. Several commercial virtual worlds have as many as nine million registered users and hundreds and thousands of servers supporting these environments.

4.3 E-Business

In e-business, scalability can be achieved by making new servers accessible as needed. For example, during a peak shopping term, more virtual servers can be made available that can cater to high shopper demand.

5. IMPLEMENTATIONS OF MAP REDUCE

Map reduce system is a major component of the cloud computing platform, and has attracted more and more attentions recently. The architecture of our implementation is similar to Hadoop, which is a typical master-worker structure. There are three roles in the system: Master, Worker and User. Master is the central controller of the system, which is in charge of data partitioning, task scheduling, load balancing and fault tolerance processing.

5.1 Implementation of Enhancement

We make three enhancements to improve the MapReduce performance in our system.

First, we treat intermediate data transfer as an independent task. Every computation task includes map and reduce subtasks. In a typical implementation such as Hadoop, reduce task starts the intermediate data transfer, which fetches the data from all the machines conducting map tasks.

Second, task scheduling is another concern on Map Reduce system, which helps to commit resources between varieties of tasks and schedule the order of task execution.

The third improvement is on data serialization. There exist frequent data exchanges between memory and disk which are generally accomplished by data serialization.

6. CONCLUSION

Cloud computing enables innovation by alleviating the need of innovators to find resources to build up, test, and make their innovations available to the user population. Innovators are free to focus on the innovation rather than the logistics of finding and managing resources that enable the innovation. In this paper, we proposed a detailed ontology for the cloud in an attempt to establish the knowledge domain of the area of cloud computing and its relevant components. We used compos ability as our methodology in constructing our cloud ontology which allowed us to capture the inter-relations between the different cloud components.

REFERENCE

[1] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peerto- peer content distribution technologies," ACM Comput. Surv., vol. 36, no. 4, pp. 335–371, 2004. | [2] M. P. Papazoglou and W.-J. Heuvel, "Service oriented architectures: approaches, technologies and research issues," The VLDB Journal, vol. 16, no. 3, pp. 389–415, 2007. | [3] X. Jin and J. Liu, "From individual based modeling to autonomy oriented computation," in Agents and Computational Autonomy, ser. Lecture Notes in Computer Science, M. N. et al., Ed., vol. 2969. Springer, 2003, pp. 151–169. | [4] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in OSDI'04, 2004. [Online]. Available: http://labs.google.com/papers/mapreduce.html | [5] "Amazon elastic compute cloud," http://aws. amazon. com/ ec2/. | [6] "Salesforce Customer Relationships Management (CRM) system," http://www.salesforce.com/. | [7] "Apex: Salesforce on-demand programming language and framework," http://developer.force.com/. | [8] "Hadoop," http://hadoop.apache.org/. | [9] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig latin: a not-so-foreign language for data processing," in SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. New York, NY, USA: ACM, 2008, pp. 1099–1110. [Online].Available: http://dx.doi.org/10.1145/1376616. 1376726 | [10] Y. Koh, R. C. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu, "An analysis of performance interference effects in virtual environments," in ISPASS. IEEE Computer Society, 2007, pp. 200–209. | [11] "Enomalism elastic computing infrastructure," http://www. enomaly.com. | [12] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," SIGOPS Oper. Syst. Rev., vol. 37, no. 5, pp. 29–43, 2003. | [13] K. Petersen, M. Spreitzer, D. Terry, and M. Theimer, "Bayou: replicated database services for world-wide applications," in EW 7: Proceedings of the 7th workshop on ACM SIGOPS European workshop. New York, NY, USA: ACM, 1996, pp. 275–280. | [14] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: amazon's highly available key-value store," in SOSP '07. New York, NY, USA: ACM, 2007, pp. 205–220. | [15] W. J. et al., "Network Communication as a Service-Oriented Capability," High Performance Computing and Grids in Action, Advances in Parallel Computing, vol. 16, March 2008. | [16] A. H. et al., "Perfsonar: A service oriented architecture for multi-domain network monitoring," in ICSOC, ser. Lecture Notes in Computer Science, B. B. et al., Ed., vol. 3826. Springer, 2005, pp. 241–254. |